

Esame di Stato Istituto Tecnico Commerciale  
Soluzione della Seconda Prova  
Indirizzo: MERCURIO  
Tema di: INFORMATICA GESTIONALE  
Anno Scolastico: 2006-2007

### Definizione dello schema concettuale del database

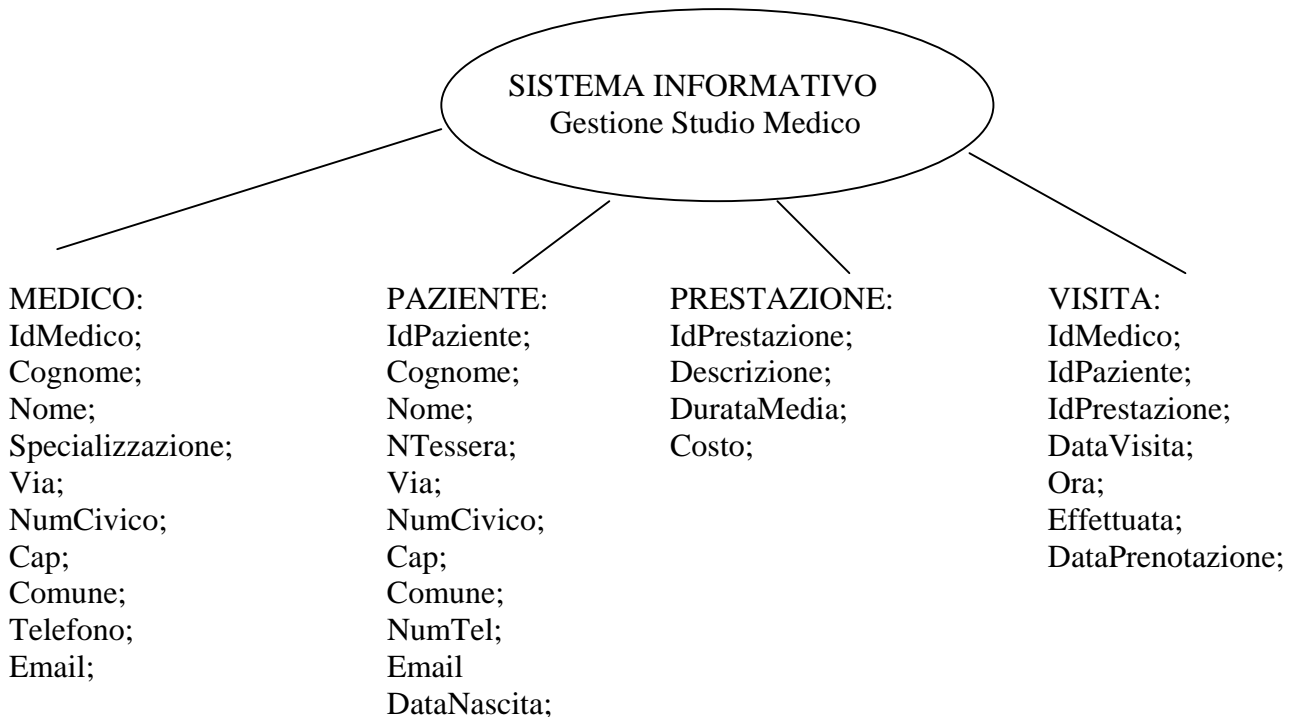
Partendo dalle specifiche contenute nel testo del problema, i vincoli di base del sistema in fase di sviluppo sono i seguenti:

- un medico può avere più visite, ma una visita è effettuata da un solo medico
- un paziente può avere effettuato visite con più medici
- le prestazioni richieste altro non siano altro che le visite stesse.
- una prestazione ha lo stesso costo a prescindere dallo specialista che la effettua.
- un medico esercita, nello studio, una sola specializzazione

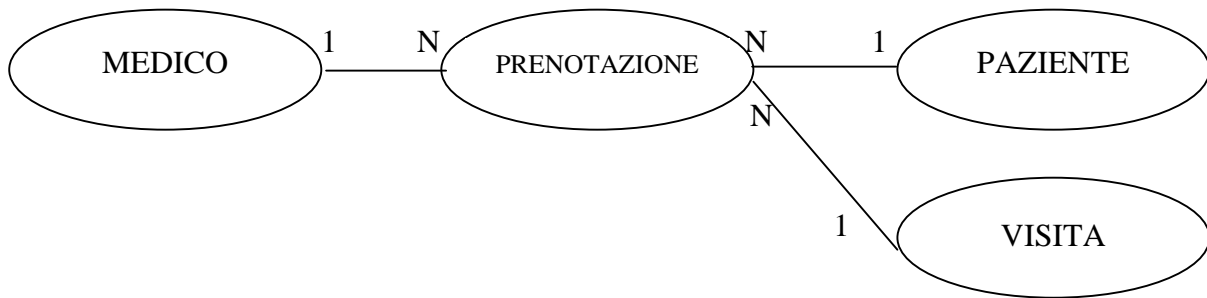
Il modello concettuale prevede le seguenti entità:

- MEDICO
- PAZIENTE
- PRESTAZIONI
- VISITE

Le entità sono individuate dagli attributi dettagliati nella figura seguente:



Le relazioni tra le diverse entità sono individuate dal seguente diagramma entità-relazioni



La relazione tra l'entità MEDICO e l'entità PRENOTAZIONE è di tipo 1:N

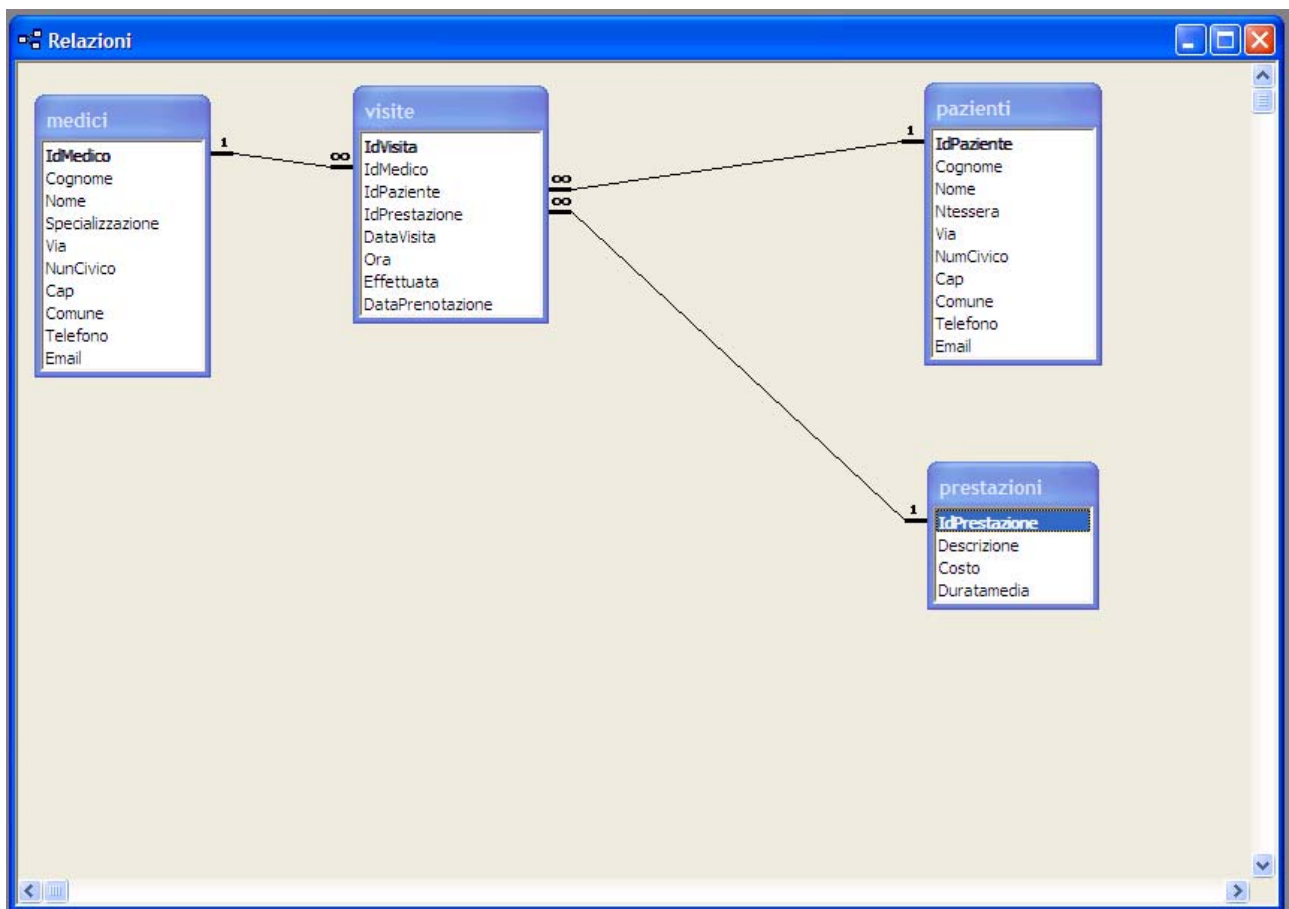
La relazione tra l'entità PAZIENTE e l'entità PRENOTAZIONE è di tipo 1:N

La relazione tra PRESTAZIONE e PRENOTAZIONE è anch'essa di tipo 1:N

Per la creazione dello schema logico della base di dati è necessario svolgere le seguenti due fasi:

- traduzione del modello concettuale in modello relazionale preliminare
- applicazione delle tre forme normali per verificare se lo schema preliminare può essere ulteriormente ottimizzato

Di seguito abbiamo lo schema logico relazionale:



Di seguito riportiamo il codice SQL interpretato da un RDBMS per la creazione del modello sopra descritto:

**CREATE TABLE Medico**

```
(IdMedico          integer,  
Cognome            char(30) not null,  
Nome               char(30) not null,  
Specializzazione  char(20),  
Via                char(20),  
NumCivico          smallint,  
Cap                integer,  
Comune             char(20),  
Telefono           char(15),  
Email              char(30),  
Primary Key (IdMedico));
```

**CREATE TABLE Paziente**

```
(IdPaziente        integer,  
Cognome            char(30) not null,  
Nome               char(30) not null,  
Ntessera           char(20),  
Via                char(20),  
NumCivico          smallint,  
Cap                integer,  
Comune             char(20),  
Telefono           char(15),  
Email              char(30),  
Primary Key (IdPaziente));
```

**CREATE TABLE Prestazione**

```
(IdPrestazione     integer,  
Descrizione        char(30) not null,  
DurataMedia        char(30) not null,  
Costo              decimal(4,2) not null,  
Primary Key (IdPrestazione));
```

**CREATE TABLE Visita**

```
(IdVisita           integer,  
IdMedico            integre not null,  
IdPaziente          integre not null,  
IdPrestazione       integre not null,  
DataVisita          date not null,  
Ora                 time not null,  
Effettuata          boolean,  
Primary Key (IdVisita),  
Foreign key (IdMedico) references Medico(IdMedico),  
Foreign key (IdPaziente) references Paziente(IdPaziente),  
Foreign key (IdPrestazione) references Prestazione(IdPrestazione));
```

## **Svolgimento delle query richieste**

Nel codice che segue i nomi racchiusi tra parentesi quadrate rappresentano parametri di input che verranno richiesti dallo RDBMS in fase di esecuzione della query.

### **1. elenco giornaliero delle visite prenotate per ogni singolo medico**

```
SELECT Medico.Cognome, Paziente.Cognome, Paziente.Nome  
FROM Medico, Visita, Paziente  
WHERE ((Visita.DataVisita)=[inserisci data: ]) AND ((Medico.IdMedico=Visita.IdMedico) AND  
(Visita.IdPaziente=Paziente.IdPaziente))  
ORDER BY Medico.Cognome;
```

### **2. elenco giornaliero delle visite prenotate e non effettuate**

```
SELECT Paziente.Cognome, Paziente.Nome, Prestazione.Descrizione  
FROM Visita, Paziente, Prestazione  
WHERE ((Visita.IdPaziente=Paziente.IdPaziente) AND  
(Visita.IdPrestazione=Prestazione.IdPrestazione) AND (Visita.DataVisita=[Inserisci la data: ]  
AND (Visita.Effettuata=False));
```

### **3. elenco settimanale contenente gli appuntamenti di ciascun medico suddivisi per giorno e per ora**

```
SELECT Medico.Cognome, Medico.Nome, Visita.DataVisita, Visita.Ora, Paziente.Cognome  
FROM Paziente, Medico, Visita  
WHERE Medico.IdMedico=Visita.IdMedico AND Paziente.IdPaziente=Visita.IdPaziente AND  
Visita.DataVisita Between [data inizio settimana: ] AND [Data fine settimana: ]  
ORDER BY Medico.Cognome, Visita.DataVisita, Visita.Ora;
```

### **4. elenco cronologico delle visite usufruite da ciascun paziente**

```
SELECT Visita.DataVisita, Paziente.Cognome, Paziente.Nome, Prestazione.Descrizione  
FROM Paziente, Visita, Prestazione  
WHERE ((Visita.Effettuata=True) AND (Paziente.IdPaziente=Visita.IdPaziente) AND  
(Prestazione.IdPrestazione=Visita.IdPrestazione))  
ORDER BY Paziente.Cognome, Visita.DataVisita;
```